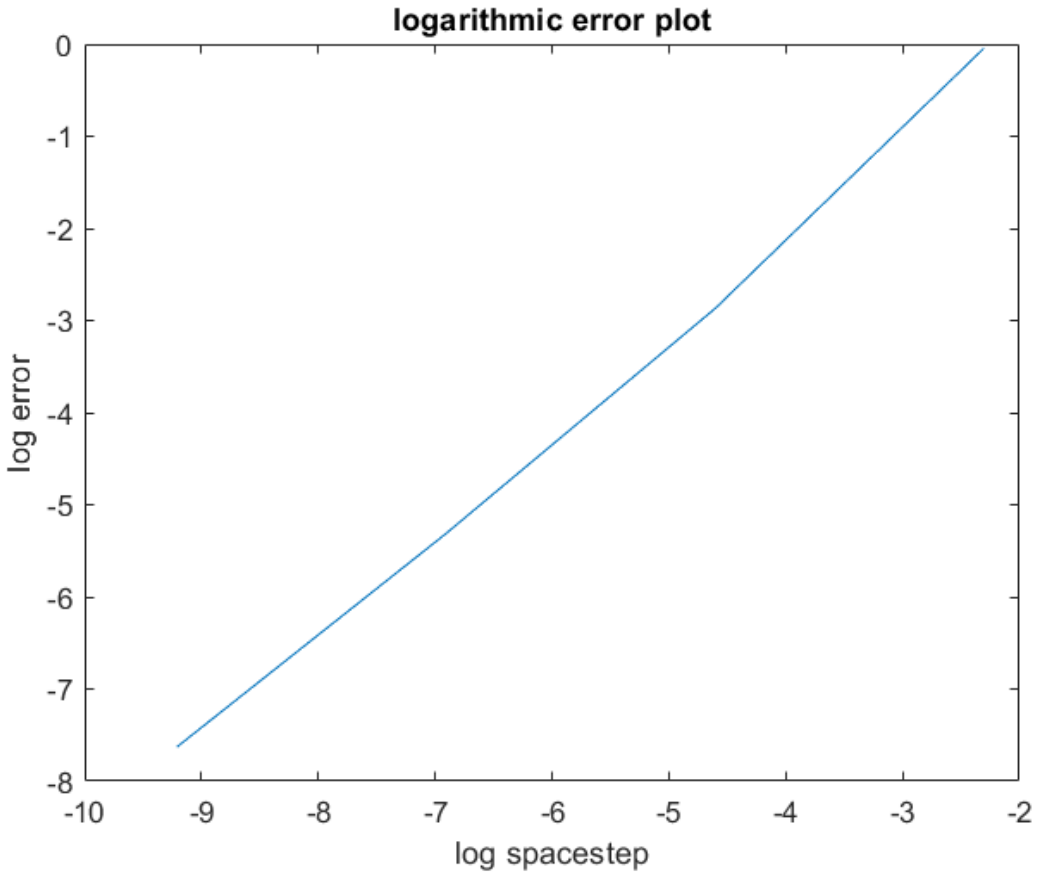
Acknowledgement: Thanks to the 16.90 TAs, I have identified the initial bug in my code. That was caused by not paying enough attention to MATLAB syntax. The initial statement was exact_u=zeros(Nx+1) and was generating a square matrix! I modified that statement into exact_u=zeros(1,Nx+1); and everything worked fine.

I have used the following code to generate the log-log plot:



As expected the slope is 1 so the method is first order accurate in space.

My code:

```
conv_velocity=1;
diffusivity=0.1;
c=1/(1-exp(conv_velocity/diffusivity));
%define Dirichlet boundary conditions
U_left_bc=1;U_right_bc=0;
dxs=[1/10, 1/100, 1/1000, 1/10000];
errs=zeros(1,4);
%errs=[1.5759,10.1571,31.6732,100.0159] values I found after running
code
for j=1:4
    dx=dxs(j);
    Nx=round(1/dx+1);
    A=zeros(Nx-1,Nx-1);
    for i=1:Nx-1
        if i==1
            A(i,i)=2*diffusivity/(dx^2);
```

```
            A(i,i+1)=conv_velocity/(2*dx)-diffusivity/(dx^2);
        elseif i==Nx-1
            A(i,i-1)=-(conv_velocity/(2*dx))-(diffusivity/(dx^2));
            A(i,i)=2*diffusivity/(dx^2);
        else
        A(i,i-1)=-(conv_velocity/(2*dx))-(diffusivity/(dx^2));
        A(i,i)=2*diffusivity/(dx^2);
        A(i,i+1)=conv_velocity/(2*dx)-diffusivity/(dx^2);
        end
    end
    b=zeros(Nx-1,1);
    b(1,1)=U_left_bc*(conv_velocity/(2*dx)+diffusivity/(dx^2));
    b(Nx-1,1)=-U_right_bc*(conv_velocity/(2*dx)-diffusivity/(dx^2));
    U_solution=A\b;
    exact_u=zeros(1,Nx+1);
    for i=1:length(exact_u)
        exact_u(i)=c*exp(i*dx*conv_velocity/diffusivity)+(1-c);
    end
       errs(j)=norm(exact_u-
[1,transpose(U_solution),0])/norm(exact_u);
end
disp(size(exact_u))
disp(size(U_solution))
%figure(1);
%plot(linspace(0,1,Nx+1),[1,transpose(U_solution),0],'-
',linspace(0,1,Nx+1),exact_u,'-')
%legend('State convection','exact');
%title('Convection');
%xlabel('location');
%ylabel('State');
disp(errs)
figure(2);
plot(log(dxs),log(errs),'-')
title('logarithmic error plot');
xlabel('log spacestep');
ylabel('log error');
```